

## Are global Context variables working on Oracle RAC?

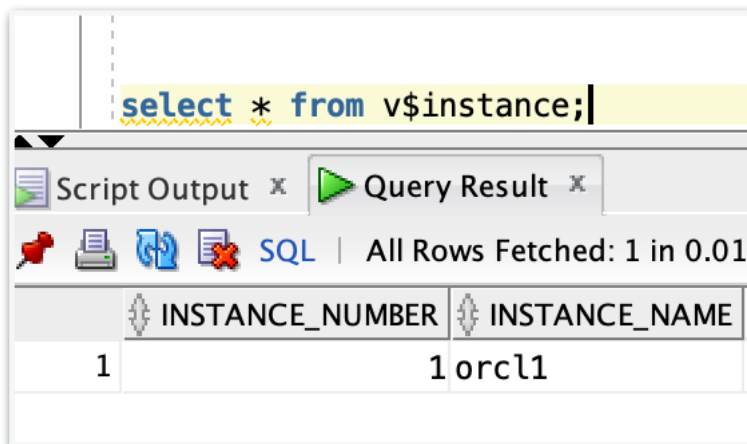
Since version 11.2 Global Application Contexts are working correctly in RAC  
This means that these global contexts are synchronised between RAC nodes.

This is a small demonstration.

---

### We are connected to RAC node 1

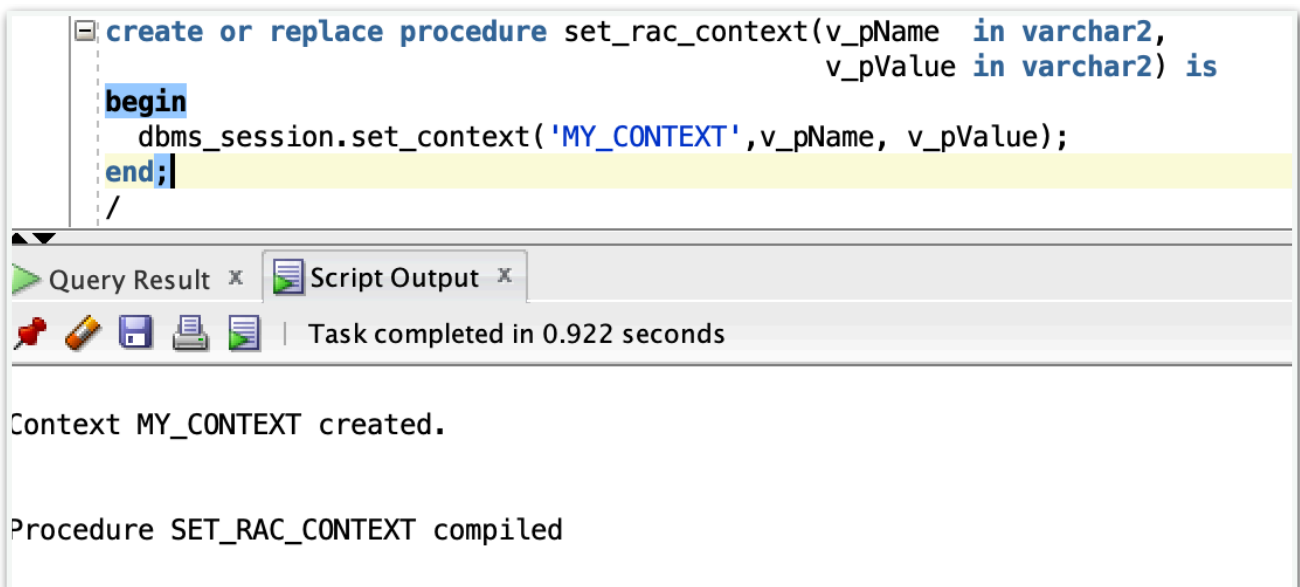
If you have two nodes the SCAN listener will automatically balance the connections.  
So there is a good chance that the second connection will land on the second node)  
It can be forced by specifying the node in the connect string (



The screenshot shows a SQL Developer window with a query editor containing the SQL statement `select * from v$instance;`. Below the editor, the 'Query Result' tab is active, displaying a table with two columns: `INSTANCE_NUMBER` and `INSTANCE_NAME`. The table contains one row with the values `1` and `1 orcl1`.

INSTANCE_NUMBER	INSTANCE_NAME
1	1 orcl1

Create the setter (the name of the procedure `set_rac_context` is just a name, and has nothing to do with RAC databases or on RAC databases)



The screenshot shows a SQL Developer window with a script editor containing the following PL/SQL code:

```
create or replace procedure set_rac_context(v_pName in varchar2,
                                           v_pValue in varchar2) is
begin
    dbms_session.set_context('MY_CONTEXT',v_pName, v_pValue);
end;
```

Below the script editor, the 'Script Output' tab is active, displaying the following output:

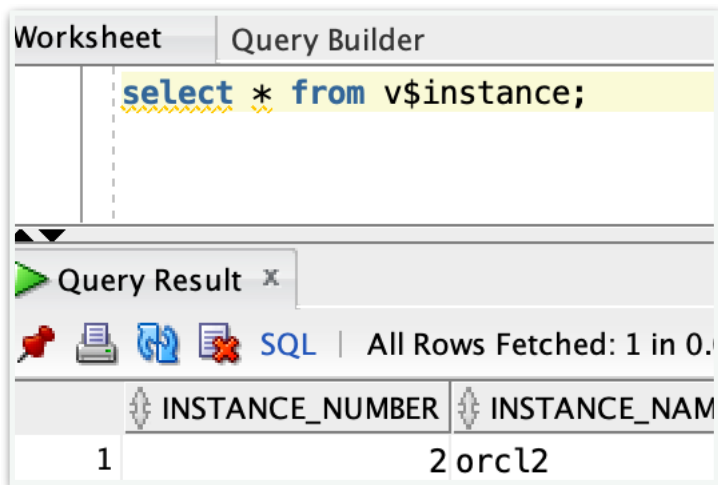
```
Context MY_CONTEXT created.

Procedure SET_RAC_CONTEXT compiled
```

Set the name value pair.

```
exec set_rac_context('Hello', 'World');
```

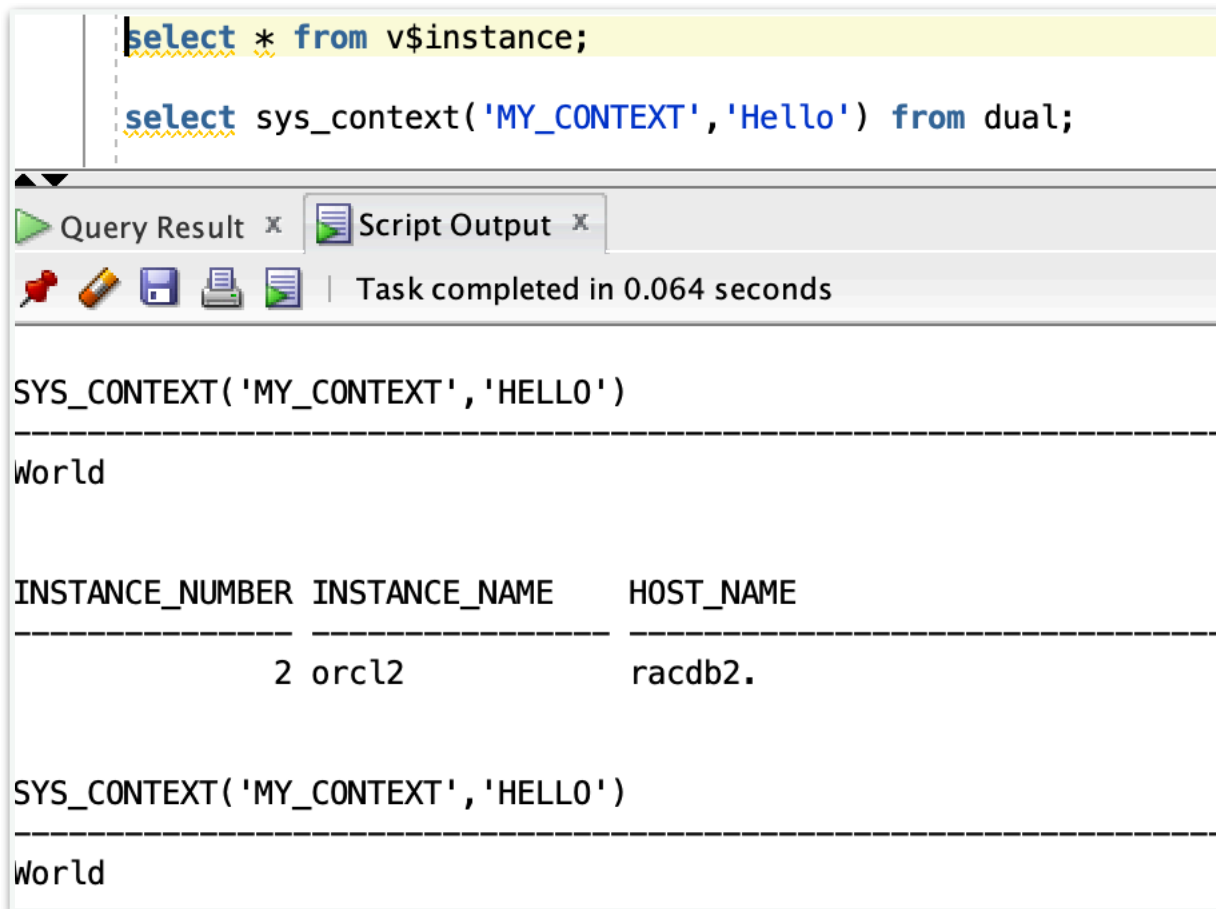
Now we are connected to instance 2.



The screenshot shows the Oracle SQL Developer interface. The top bar has 'Worksheet' and 'Query Builder' tabs. The main area contains the SQL query: `select * from v$instance;`. Below the query is a 'Query Result' window with a toolbar and the text 'All Rows Fetched: 1 in 0.'. The result is displayed in a table with two columns: 'INSTANCE\_NUMBER' and 'INSTANCE\_NAME'. The first row shows the value '2' for the instance number and 'orcl2' for the instance name.

INSTANCE_NUMBER	INSTANCE_NAME
2	orcl2

And we read back the value on node 2.



The screenshot shows the Oracle SQL Developer interface with two queries. The first query is `select * from v$instance;`. The second query is `select sys_context('MY_CONTEXT', 'Hello') from dual;`. The results are displayed in a 'Query Result' window with a toolbar and the text 'Task completed in 0.064 seconds'. The first result is a table with three columns: 'INSTANCE\_NUMBER', 'INSTANCE\_NAME', and 'HOST\_NAME'. The second result is the text 'World'.

INSTANCE_NUMBER	INSTANCE_NAME	HOST_NAME
2	orcl2	racdb2.

SYS\_CONTEXT('MY\_CONTEXT', 'HELLO')

World

SYS\_CONTEXT('MY\_CONTEXT', 'HELLO')

World

It also works in the other direction.

```
select * from v$instance;
```

```
select sys_context('MY_CONTEXT','Hello') from dual;
```

Query Result x Script Output x

Task completed in 0.064 seconds

SYS\_CONTEXT('MY\_CONTEXT','HELLO')

---

World

INSTANCE_NUMBER	INSTANCE_NAME	HOST_NAME
2	orcl2	racdb2.

SYS\_CONTEXT('MY\_CONTEXT','HELLO')

---

World

Now we set the variable to Underworld

```
exec set_rac_context('Hello','Underworld');
```

```
select sys_context('MY_CONTEXT','Hello') from dual;
```

Script Output x Query Result x

All Rows Fetched: 1 in 0.013 seconds

SYS_CONTEXT('MY_CONTEXT','HELLO')
1 Underworld

done

Back on node 1.

The screenshot shows a SQL IDE interface. At the top, two SQL queries are entered in a text area:

```
select sys_context('MY_CONTEXT','Hello') from dual;  
select * from v$instance;
```

Below the text area is a toolbar with icons for script output, query result, and task completion. The status bar indicates "Task completed in 0.071 seconds".

The output area displays the results of the queries:

PL/SQL procedure successfully completed.

SYS\_CONTEXT('MY\_CONTEXT','HELLO')

-----

Underworld

-----

INSTANCE_NUMBER	INSTANCE_NAME	HOST_NAME
1	orcl1	racdb1.

Its magic